FMT_engraftment

Code ▾

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Hide

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
getwd()
```

```
[1] "C:/work/fmt_enterotype/a_microbiome/analysis"
```

Hide

```
source("pre_processing.R")
```

```
[1] 0.01
[1] 0.01
[1] 0.01
[1] 0.01
```

Hide

```r
##engraftment
engraft_ratio <- function(L6_abundance, FMT_config){
    L6_abundance = L6_abundance + 1e-5
    meta_fil_config1_sub <- FMT_config[,c("After", "Before", "Donor")]

    one_fold_change <- apply(meta_fil_config1_sub, 1, function(x, abundance){
        index <- match(x, colnames(abundance))
        index_L6 <- abundance[,index]

        index_fc<- apply(index_L6, 1, function(y, x){
            out_p <- log((y[1]/y[2])+1);out_d <- log((y[1]/y[3])+1);c(out_p, out_d, y[1])}, x=
x)

        quadrant1 <- mean(index_fc[3, index_fc[1,] > index_fc[2,]], trim=0.05)*length(index_fc[
3,index_fc[1,] > index_fc[2,]])

        quadrant2 <- mean(index_fc[3, index_fc[1,] < index_fc[2,]], trim=0.05)*length(index_fc[
3,index_fc[1,] < index_fc[2,]])

        -log(quadrant2/quadrant1)
    }, L6_abundance)

    fc_coin <- as.data.frame(cbind(as.numeric(one_fold_change), (FMT_config$postfmt_symptoms),
 meta_fil_config1_sub[,'Before']))
    colnames(fc_coin)<- c('fc', 'group', 'before')
    rownames(fc_coin) <- (FMT_config$SRA_Sample)
    fc_coin$fc <- as.numeric(as.character(fc_coin$fc))

    # prj_list <- fc_coin$prj
    # list <- NULL
    # for(i in unique(prj_list)){
    #     if (length(prj_list[prj_list %in% i]) > 2){
    #         list <- c(list, i)
    #     }
    # }
    # fc_coin_f <- fc_coin[fc_coin$prj %in% list,]

    pval<-wilcox_test(fc ~ group , fc_coin)#| prj
    pval
    return(list( 'fc_ratio'=fc_coin, 'ratio_p'=pval))

}
```

Hide

```
###meta_fil_config1
###L6_rela_fil_sAg_remove
library(coin)
L6_rela_fil_sAg_remove_a1 <- L6_rela_fil_sAg_remove + 1
seq(0.1, 0.9, 0.05) -> quan


quntile_fold_change <- function(L6_rela_fil_sAg_remove_a1, meta_fil_config1, pre_don, quan){
#"Previous_sra"
    sid_SRA_Sample <- meta_fil_config1$SRA_Sample
    sid_pre_don<- meta_fil_config1[, pre_don]
    # colnames(L6_rela_fil_sAg_remove_a1)
    index_SRA_Sample <- match(sid_SRA_Sample, colnames(L6_rela_fil_sAg_remove_a1))
    index_pre_don <- match(sid_pre_don, colnames(L6_rela_fil_sAg_remove_a1))
    quntile_fold_change <- apply(L6_rela_fil_sAg_remove_a1, 1, function(x, index_SRA_Sample, index_pre_don){
        mean_fold<- mean(log2((quantile((x[index_SRA_Sample]), quan) / quantile((x[index_pre_don]), quan)) + 0));
        c(mean_fold, mean(x[index_SRA_Sample], trim=0.05))
    }, index_SRA_Sample=index_SRA_Sample, index_pre_don=index_pre_don)
    t(quntile_fold_change)
}
```

Hide

```
engra_h2r <- function(L6_rela_fil_sAg, meta_fil_config, class){

  if(length(class) == 3){
    c_meta <- meta_fil_config
  }else{
    # pre_c = as.character(plotdata[plotdata[,"data.cluster"] %in% class, "sample"])
    #pre_c_psym = as.character(plotdata[plotdata[,"data.cluster"]  class, "group"])
    c_meta = meta_fil_config#[meta_fil_config[,"Previous_sra"] %in% pre_c, ]
    c_meta<-c_meta[c_meta$PRJ %in% names(table(c_meta[,"PRJ"]))[table(c_meta[,"PRJ"]) > 2],]
    c_meta$PRJ <- as.factor(c_meta$PRJ)
  }

  L6_rela_fil_sAg[,unique(as.character(c_meta[, "SRA_Sample"]))] -> after_genus

  nrow(after_genus)->after_row
  rank_after<-apply(after_genus, 2, function(x){
    tmp_rank <- rank(x, ties.method = c("min"))/length(x)
    tmp_rank <- floor(floor(tmp_rank * 10)/2)*0.2
    tmp_rank[tmp_rank < 0.10] = 0.05
    tmp_rank
    #rank(after_genus[1,], ties.method = c("min"))/length(after_genus[1,])
  })
  rowMeans(rank_after)->after_mean_rank
  seq(0.1, 0.9, 0.1) -> quan

  L6_rela_fil_sAg1 <- as.data.frame(L6_rela_fil_sAg)
  L6_rela_fil_sAg1$rowname = 0
  L6_rela_fil_sAg1$rowname <- as.character(rownames(L6_rela_fil_sAg1))
  rows <- nrow(c_meta)
  engra <- apply(L6_rela_fil_sAg1, 1, function(x){
    fmt_dat <- matrix(0, nrow = rows, ncol = 3)
    #print(names(x))
    #fmt_dat <-cbind(x[meta_fil_config$Previous_sra], x[meta_fil_config$SRA_Sample], x[meta_fil_config$Donor_sra])
    fmt_dat[,1] <- as.numeric(x[as.character(c_meta$SRA_Sample)])
    fmt_dat[,2] <- as.numeric(x[as.character(c_meta$Donor_sra)])
    fmt_dat[,3] <- as.numeric(x[as.character(c_meta$Previous_sra)])

    pt<-as.data.frame(cbind(c(fmt_dat[,1], fmt_dat[,3]), rep(c('1', '2'), each=rows), c(c_meta[, "PRJ"], c_meta[, "PRJ"])))
    colnames(pt)<- c('ab', 'group', 'prj')
    pt$ab <- as.numeric(as.character(pt$ab))
    tmp_test <- wilcox_test(ab ~ group | prj, pt)
    pval_a <- NA
    pval_a <- pvalue(tmp_test)

    colnames(fmt_dat) <- c('p', 'a', 'd')
    as.data.frame(fmt_dat)->fmt_dat
    coff <- as.numeric(after_mean_rank[x['rowname']])

    means <- rep(mean(fmt_dat[,'a']), nrow(fmt_dat))
    fmt_dat <- fmt_dat / 1e5 + 1e-6
    lm_test <- lm(a ~  p + d,data=fmt_dat)
    lm_var <- anova(lm_test)[,2]
    var_sum <- sum(lm_var)
    if(var_sum == 0){return(c(simp_names(x['rowname']), 0, 0, 0, coff, 0, pval_a))}
    h2r <- (lm_var[1] + lm_var[2])/var_sum
    h2r_p <- lm_var[1]/var_sum
    h2r_d <- lm_var[2]/var_sum
    kmh2r<- as.numeric(coff * h2r)



    # glm_var <- tryCatch({
    #   glm_test <- glm(a ~ p + d,data=fmt_dat, family=quasipoisson(link='probit'))
    #   anova(glm_test)[,4]
    # }, error = function(e) {
    #   return(0)
    # }, finally = {})
    # # glm_var <- anova(glm_test)[,4]
    # var_sum <- sum(glm_var)
    # if(var_sum == 0){return(c(simp_names(x['rowname']), 0, 0, 0, coff, 0, pval_a))}
    # h2r <- (glm_var[2] + glm_var[3])/var_sum
    # h2r_p <- glm_var[2]/var_sum
    # h2r_d <- glm_var[3]/var_sum
    # kmh2r<- as.numeric(coff * h2r)
    c(simp_names(x['rowname']), h2r_p, h2r_d, h2r, coff, kmh2r, pval_a)
  })

  engra <- t(engra)
  colnames(engra)<-c( 'id', 'h2r_p', 'h2r_d', 'h2r', 'coff', 'kmh2r', 'pval')
  h2r_res <- data.frame(engra)
  h2r_res$h2r_d <- as.numeric(as.character(h2r_res$h2r_d))
  h2r_res$kmh2r <- as.numeric(as.character(h2r_res$kmh2r))
  h2r_res$h2r <- as.numeric(as.character(h2r_res$h2r))
  h2r_res$pval <- as.numeric(as.character(h2r_res$pval))

  diff_qvalue <- p.adjust(h2r_res$pval, method='fdr')
  h2r_res <- cbind(h2r_res, diff_qvalue)
  nrow(h2r_res) -> row
  return(h2r_res)
}
```

Hide

```
cal_partition<-function(pre_don_fc_da){
    quadrant1 <- sum(pre_don_fc_da[pre_don_fc_da$pre_fc > pre_don_fc_da$don_fc, c("pre_ab")])/100000
    quadrant2 <- sum(pre_don_fc_da[pre_don_fc_da$pre_fc < pre_don_fc_da$don_fc, c("pre_ab")])/100000
    # quadrant1 <- sum(pre_don_fc_da[pre_don_fc_da$pre_fc > 0 & pre_don_fc_da$don_fc > 0, c("pre_ab")])/100000
    # quadrant2 <- sum(pre_don_fc_da[pre_don_fc_da$pre_fc < 0 & pre_don_fc_da$don_fc > 0, c("pre_ab")])/100000
    # quadrant3 <- sum(pre_don_fc_da[pre_don_fc_da$pre_fc < 0 & pre_don_fc_da$don_fc < 0, c("pre_ab")])/100000
    # quadrant4 <- sum(pre_don_fc_da[pre_don_fc_da$pre_fc > 0 & pre_don_fc_da$don_fc < 0, c("pre_ab")])/100000

    return(c(quadrant1, quadrant2))#, quadrant3, quadrant4))
}
```

Hide

```
###ratio distrubution
one_fold_change <- function(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na){

    meta_fil_config1_sub <- meta_fil_config1_na[,c("SRA_Sample", "Previous_sra", "Donor_sra")]
    one_fold_change <- apply(meta_fil_config1_sub, 1, function(x, L6_rela_fil_sAg_remove_a1){
        index <- match(x, colnames(L6_rela_fil_sAg_remove_a1))
        index_L6 <- L6_rela_fil_sAg_remove_a1[,index]

        index_fc<- apply(index_L6, 1, function(y, x){
            out_p <- log((y[1]/y[2])+1);out_d <- log((y[1]/y[3])+1);c(out_p, out_d, y[1])}, x=
x)

        quadrant1 <- mean(index_fc[3, index_fc[1,] > index_fc[2,]], trim=0.05)*length(index_fc[
3,index_fc[1,] > index_fc[2,]])##(index_fc[1, index_fc[1,] > index_fc[2,]] - index_fc[2, index_
fc[1,] > index_fc[2,]])*
        quadrant2 <- mean(index_fc[3, index_fc[1,] < index_fc[2,]], trim=0.05)*length(index_fc[
3,index_fc[1,] < index_fc[2,]])#(index_fc[2, index_fc[1,] < index_fc[2,]] - index_fc[1, index_f
c[1,] < index_fc[2,]])
        -log(quadrant2/quadrant1)
    }, L6_rela_fil_sAg_remove_a1)

    fc_coin <- as.data.frame(cbind(as.numeric(one_fold_change), (meta_fil_config1_na$postfmt_sy
mptoms), (meta_fil_config1_na$PRJ)))
    colnames(fc_coin)<- c('fc', 'group', 'prj')
    rownames(fc_coin) <- (meta_fil_config1_na$SRA_Sample)
    fc_coin$fc <- as.numeric(as.character(fc_coin$fc))

    prj_list <- fc_coin$prj
    list <- NULL
    for(i in unique(prj_list)){
        if (length(prj_list[prj_list %in% i]) > 2){
            list <- c(list, i)
        }
    }
    fc_coin_f <- fc_coin[fc_coin$prj %in% list,]

    pval<-wilcox_test(fc ~ group | prj, fc_coin_f)
    pval
    return(list('fc_ratio'=fc_coin, 'ratio_p'=pval))

}
```

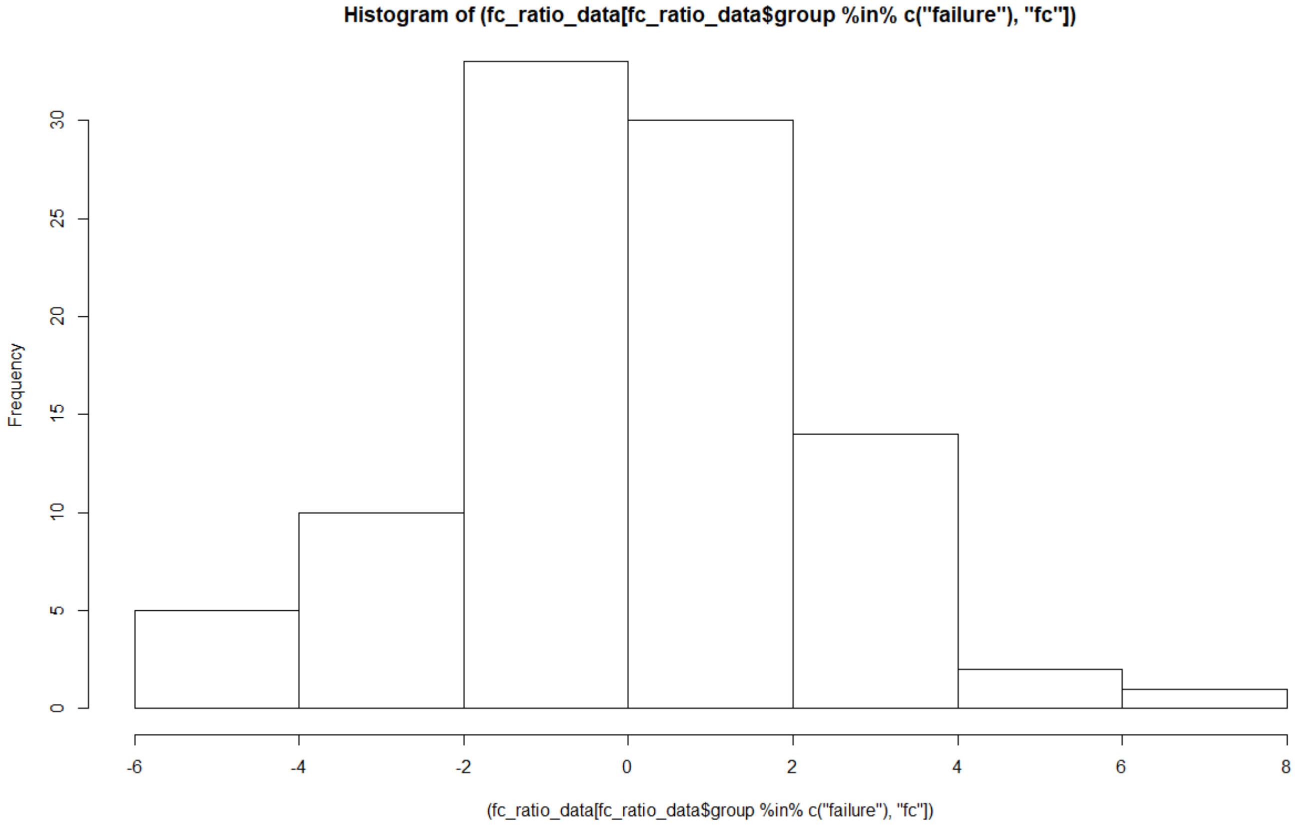
Hide

```
meta_fil_config1_na <-meta_fil_config1[!meta_fil_config1$postfmt_symptoms %in% c(NA),]
out_fc_ratio <- one_fold_change(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na)

out_fc_ratio$ratio_p
```

```


    Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (failure, response)
     stratified by prj
Z = -2.8381, p-value = 0.004538
alternative hypothesis: true mu is not equal to 0
```

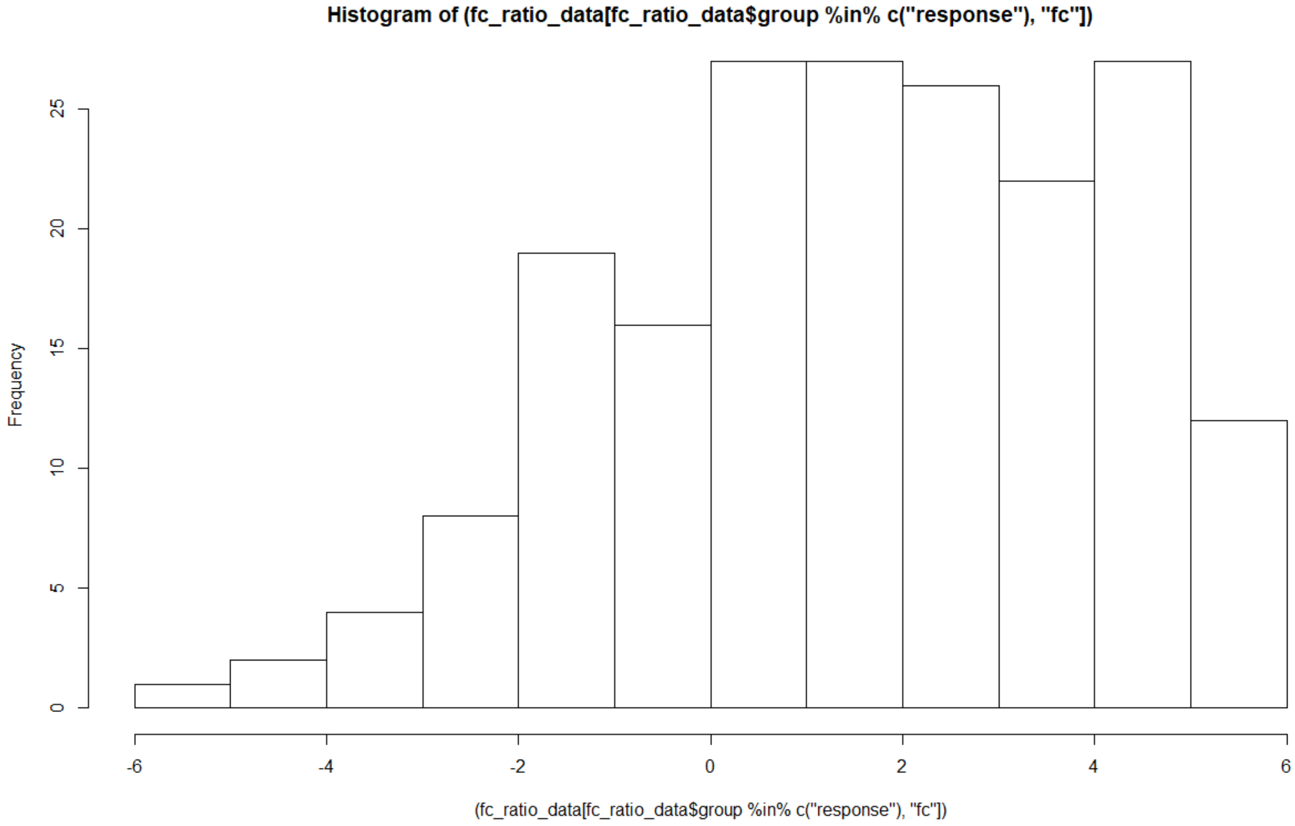
Hide

```
fc_ratio_data <- out_fc_ratio$fc_ratio
hist((fc_ratio_data[fc_ratio_data$group %in% c('failure'), 'fc']))
```



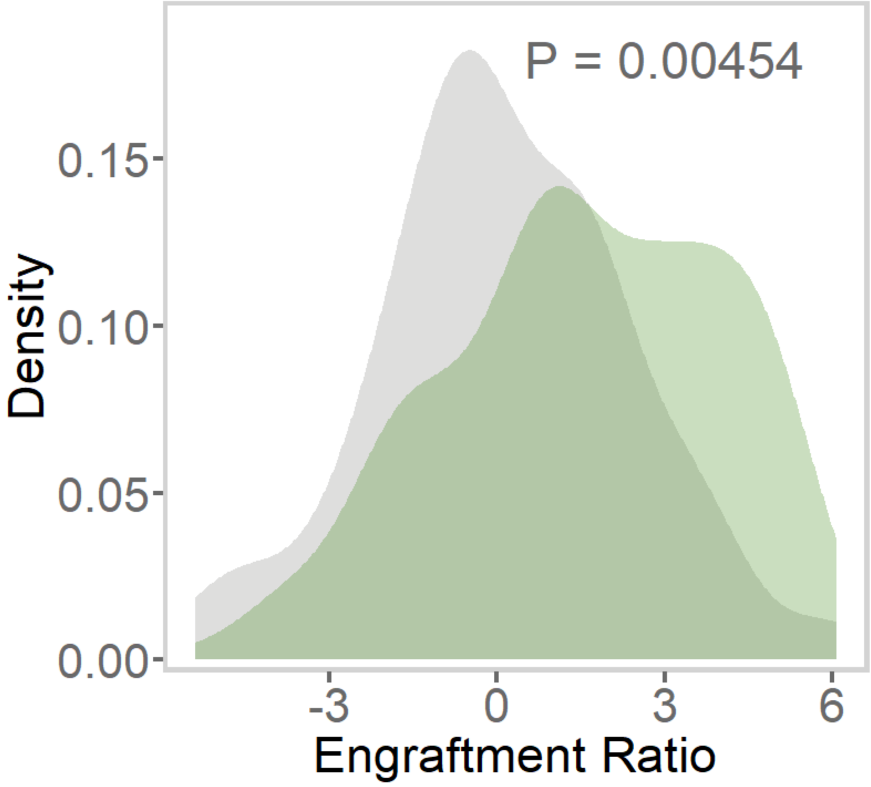
Hide

```
hist((fc_ratio_data[fc_ratio_data$group %in% c('response'), 'fc']))
```



Hide

```
library(ggpubr)
# ggboxplot(fc_ratio_data,x='group', y='fc', fill='white', alpha = 0, size=0)+
cex=1.5
ggplot()+theme_classic()+
    geom_density(aes(fc, stat(density), fill = group), fc_ratio_data, alpha = (0.3), color='NA'
)+
    scale_y_continuous(expand = expansion(mult =c(0.02, 0.08)))+#, limits = c(-0.005, 0.17)
    # stat_compare_means(comparisons = list(c('failure', 'response')), method = 'wilcox.test',
 label = "p.signif", label.y = 1, size=8*cex)+
    scale_fill_manual(values = c("#90908D", "#4D9127"))+
    scale_alpha_manual(values=c(1, 0.5))+
  # theme(text=element_text(family ="sans", size=24*cex), plot.title = element_text(size=24.6*c
ex, hjust = 0.5), axis.text = element_text(size=21*cex, color ='dimgray'), axis.title=element_t
ext(family ="sans", size=24*cex))+
  theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hjust =
0.5), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = element_text(size=34
), axis.title.y = element_text(size=34), axis.ticks = element_line(size=1.5, color ='dimgray'),
axis.ticks.length = unit(7, "pt"))+
  theme(legend.position = "right")+xlab(label = 'Engraftment Ratio')+ylab("Density")+labs(title
='')+
  geom_text(aes(x=3, y=0.18, label=paste('P =', format(round(pvalue(out_fc_ratio$ratio_p), 5),
 nsmall = 4) )), size=8*cex, color='dimgrey')+
  theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(5, 0.6)
        ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
        ,axis.line=element_line(colour=NA, size = 0), axis.ticks = element_line(size=1.5, color
='dimgray'), axis.ticks.length = unit(7, "pt")
        ,legend.key = element_rect(fill = NA, color = NA))+
  # theme(axis.text.y = element_blank(), axis.ticks.y = element_blank(), axis.title.y = element
_blank(),)+
  guides(colour = guide_legend(override.aes = list(size=3)));
```



Hide

```
fig3i = 1
ggsave(paste("./figure3/3main_all", fig3i, ".pdf", sep = ''), device = "pdf")
```

```
Saving 12.9 x 8 in image
```

Hide

```
fig3i = fig3i + 1
# ggdensity(fc_ratio_data, x='fc', fill='group', palette = c("#90908D", "#4D9127"), alpha = c(0.3), y = "..density..")
    # stat_compare_means(comparisons = list(c('failure', 'response')), method = 'wilcox.test', label = "p.signif", size=8)
```

Hide

```
meta_fil_config1_na <-meta_fil_config1[!meta_fil_config1$postfmt_symptoms %in% c(NA),]
out_fc_ratio_1 <- one_fold_change(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c('before1'),])
out_fc_ratio_1$ratio_p
```

```


    Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (failure, response)
     stratified by prj
Z = -3.7088, p-value = 0.0002082
alternative hypothesis: true mu is not equal to 0
```

Hide

```
out_fc_ratio <- one_fold_change(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c('before2'),])
out_fc_ratio$ratio_p
```

```


    Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (failure, response)
     stratified by prj
Z = -0.38067, p-value = 0.7034
alternative hypothesis: true mu is not equal to 0
```

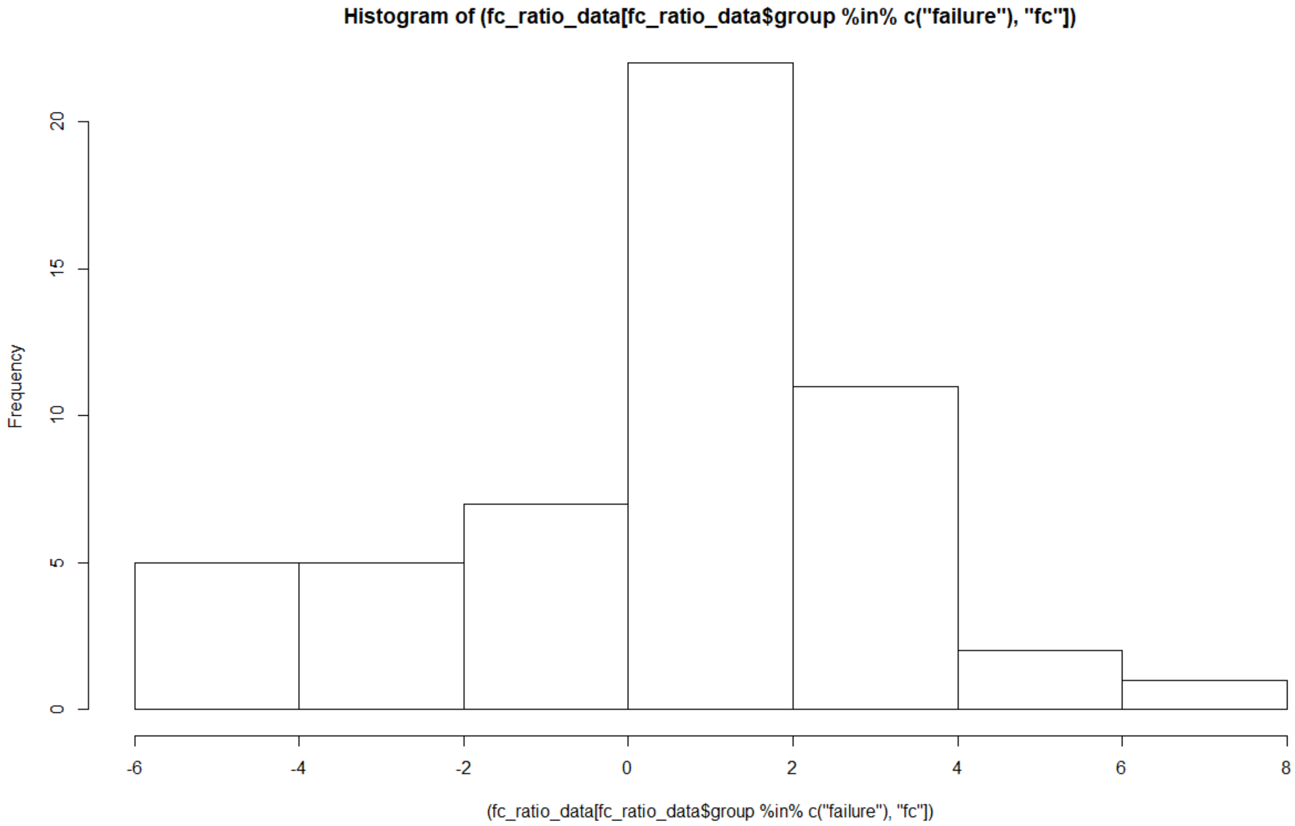
Hide

```
out_fc_ratio <- one_fold_change(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c('before1'),])
out_fc_ratio$ratio_p
```

```

	Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (failure, response)
	 stratified by prj
Z = -3.7088, p-value = 0.0002082
alternative hypothesis: true mu is not equal to 0
```

Hide

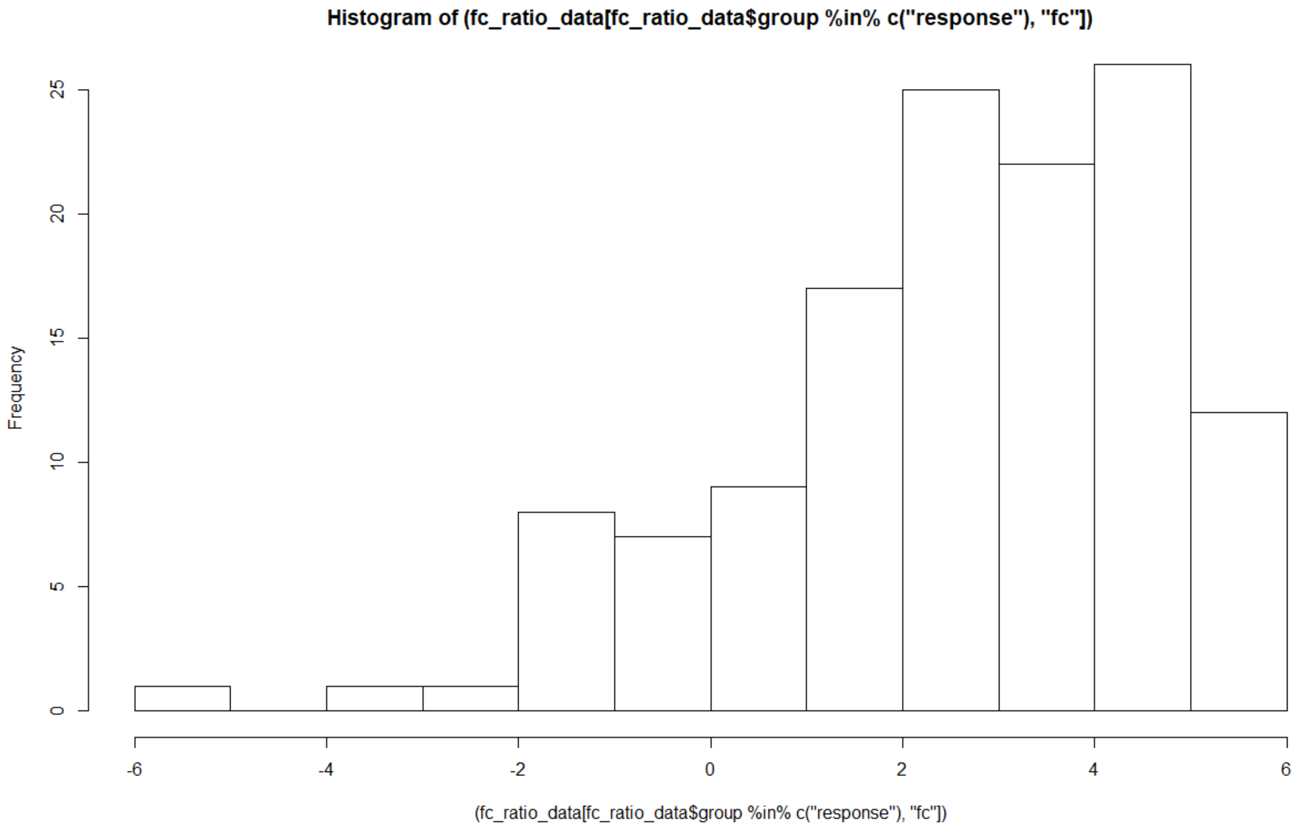
```
fc_ratio_data <- out_fc_ratio$fc_ratio
hist((fc_ratio_data[fc_ratio_data$group %in% c('failure'), 'fc']))
```



Histogram of (fc_ratio_data[fc_ratio_data$group %in% c("failure"), "fc"])

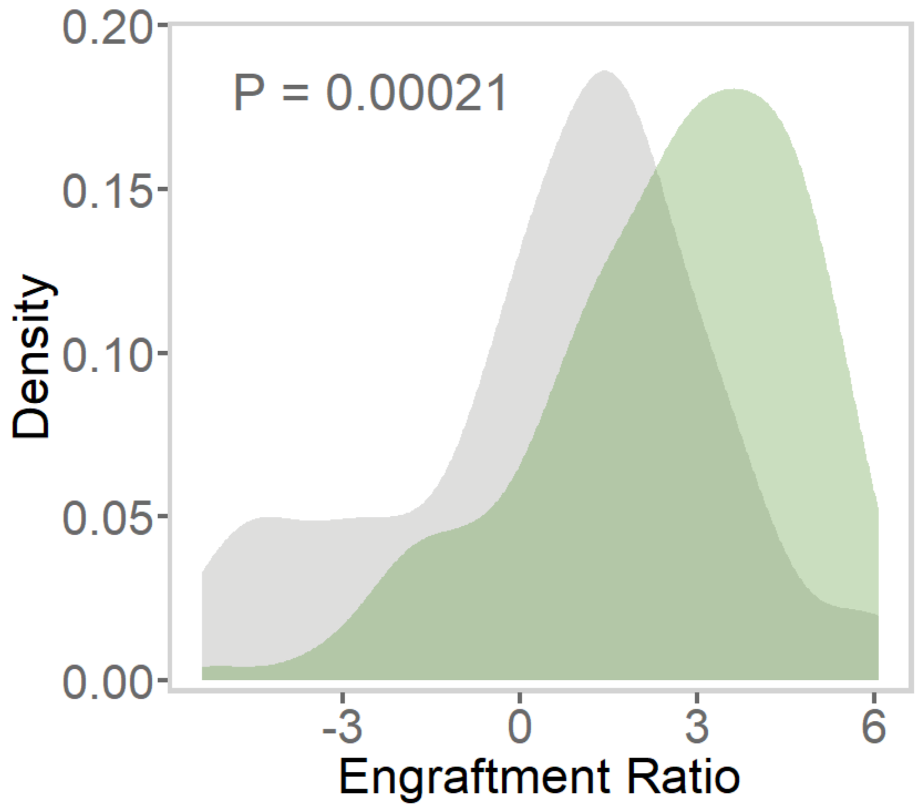
Hide

```
hist((fc_ratio_data[fc_ratio_data$group %in% c('response'), 'fc']))
```



Hide

```
library(ggpubr)
# ggboxplot(fc_ratio_data,x='group', y='fc', fill='white', alpha = 0, size=0)+
cex=1.5
ggplot()+theme_classic()+
    geom_density(aes(fc, stat(density), fill = group), fc_ratio_data, alpha = (0.3), color='NA'
)+
    scale_y_continuous(expand = expansion(mult =c(0.02, 0.08)))+#, limits = c(-0.005, 0.17)
    # stat_compare_means(comparisons = list(c('failure', 'response')), method = 'wilcox.test',
 label = "p.signif", label.y = 1, size=8)+
    scale_fill_manual(values = c("#90908D", "#4D9127"))+
    scale_alpha_manual(values=c(1, 0.5))+
  # theme(text=element_text(family ="sans", size=24*cex), plot.title = element_text(size=24*ce
x, hjust = 0.5), axis.text = element_text(size=21*cex, color ='dimgray'))+
  theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hjust =
0.5), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = element_text(size=34
), axis.title.y = element_text(size=34))+
  theme(legend.position = "right")+xlab(label = 'Engraftment Ratio')+ylab("Density")+labs(title
='')+
  geom_text(aes(x=-2.5, y=0.18, label=paste('P =', format(round(pvalue(out_fc_ratio$ratio_p), 5
), nsmall = 4) )), size=8*cex, color='dimgrey')+
  theme(aspect.ratio = 0.9, legend.background=element_blank(), legend.position=c(5, 0.6)
        ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
        ,axis.line=element_line(colour=NA, size = 0), , axis.ticks = element_line(size=1.5, col
or ='dimgray'), axis.ticks.length = unit(7, "pt")
        ,legend.key = element_rect(fill = NA, color = NA))+
  # theme(axis.text.y = element_blank(), axis.ticks.y = element_blank(), axis.title.y = element
_blank(),)+
  guides(colour = guide_legend(override.aes = list(size=3)));
```



Hide

```
ggsave(paste("./figure4/4main_engraft_e", fig3i, ".pdf", sep = ''), device = "pdf")
```

```
Saving 12.9 x 8 in image
```

Hide

```
fig3i = fig3i + 1
```

Hide

```
meta_fil_config1_na <-meta_fil_config1[!meta_fil_config1$postfmt_symptoms %in% c(NA),]
out_fc_ratio_1 <- one_fold_change(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c('before1') & meta_fil_config1_na$Diease1 %in% c('CDI'),])
out_fc_ratio_1$ratio_p
```

```

	Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (failure, response)
	 stratified by prj
Z = -3.5788, p-value = 0.0003451
alternative hypothesis: true mu is not equal to 0
```

Hide

```
out_fc_ratio <- one_fold_change(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c('before1') & meta_fil_config1_na$Diease1 %in% c('UC', 'CD'),])
out_fc_ratio$ratio_p
```

```

    Asymptotic Wilcoxon-Mann-Whitney Test

data:  fc by group (failure, response)
     stratified by prj
Z = -0.6333, p-value = 0.5265
alternative hypothesis: true mu is not equal to 0
```

Hide

```
out_fc_ratio_all <- one_fold_change(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c('before1', 'before2'),])

out_fc_ratio_all_dat <- out_fc_ratio_all$fc_ratio
```

Hide

```
set.seed(666)
library(knitr)
knit('engraft_validation_feast.Rmd', tangle=TRUE)
```

```
[1] "engraft_validation_feast.R"
```

Hide

```
source('engraft_validation_feast.R')
```

```
[1] "Calculating mixinig proportions for sink 10"
[1] "Calculating mixinig proportions for sink 20"
[1] "Calculating mixinig proportions for sink 30"
[1] "Calculating mixinig proportions for sink 40"
[1] "Calculating mixinig proportions for sink 50"
[1] "Calculating mixinig proportions for sink 60"
[1] "Calculating mixinig proportions for sink 70"
[1] "Calculating mixinig proportions for sink 80"
[1] "Calculating mixinig proportions for sink 90"
[1] "Calculating mixinig proportions for sink 100"
[1] "Calculating mixinig proportions for sink 110"
[1] "Calculating mixinig proportions for sink 120"
[1] "Calculating mixinig proportions for sink 130"
[1] "Calculating mixinig proportions for sink 140"
[1] "Calculating mixinig proportions for sink 150"
[1] "Calculating mixinig proportions for sink 160"
[1] "Calculating mixinig proportions for sink 170"
[1] "Calculating mixinig proportions for sink 180"
[1] "Calculating mixinig proportions for sink 190"
[1] "Calculating mixinig proportions for sink 200"
[1] "Calculating mixinig proportions for sink 210"
[1] "Calculating mixinig proportions for sink 220"
[1] "Calculating mixinig proportions for sink 230"
[1] "Calculating mixinig proportions for sink 240"
[1] "Calculating mixinig proportions for sink 250"
[1] "Calculating mixinig proportions for sink 260"
[1] "Calculating mixinig proportions for sink 270"
[1] "Calculating mixinig proportions for sink 280"
[1] "Calculating mixinig proportions for sink 286"
```

Hide

```
##validation FEAST
#out_fc_ratio_all_dat
#feast_engra

validation_engra_feast <- merge(out_fc_ratio_all_dat, feast_engra, by="row.names")
dim(validation_engra_feast)
```

```
[1] 286   8
```

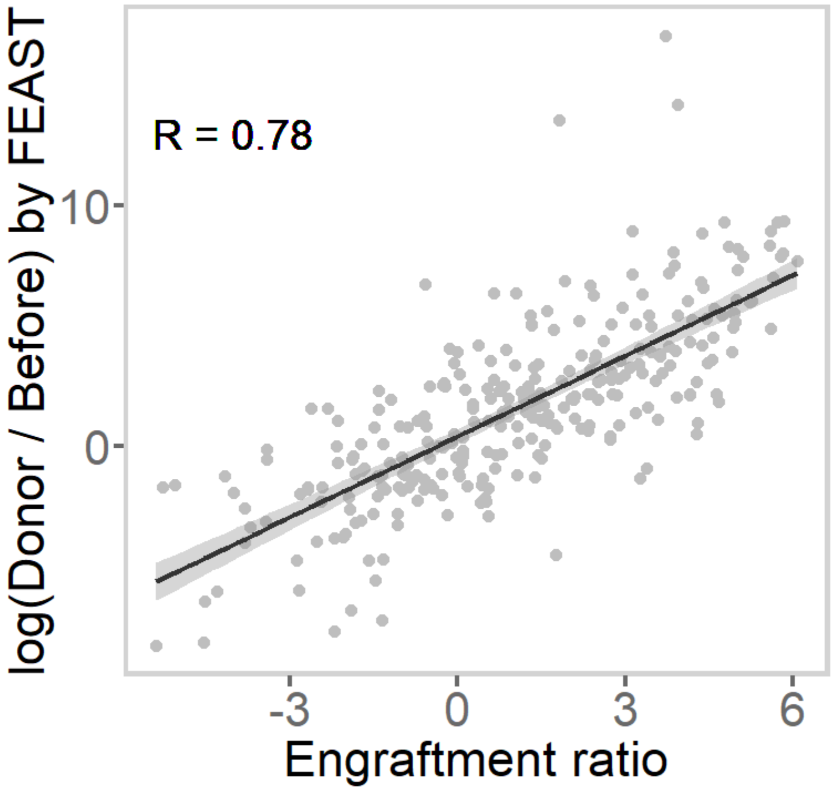
Hide

```
cor_val <- cor(validation_engra_feast$fc, validation_engra_feast$ratio, method = 'spearman')

cor_val
```

```
[1] 0.7821824
```

Hide

```
# cairo_pdf(filename = "./figure3/feast.pdf")

ggplot(validation_engra_feast, aes(fc, ratio))+geom_point(color ='grey', size=3.5)+
    geom_smooth(method = 'lm', formula = y ~ x, size=1.5, color='grey20')+
    labs(x= c('Engraftment ratio'), y=c('log(Donor / Before) by FEAST'), title = c(''))+
    geom_text(aes(x=-4, y=13, label=paste('R = ', round(cor_val, 2), sep = '')), size=10)+
    # theme(plot.title = element_text(size=24, hjust = 0.5), axis.text = element_text(size=21,
 color ='dimgray'), axis.title.y = element_text(size=30, hjust = 0.5, vjust = 1), axis.title.x
 =element_text(size=30),
          # title=element_text(family ="sans", size=21),
            # text=element_text(family ="sans", size=22),aspect.ratio=0.95)+
            #text=element_text(family ="sans Neue", size=21), plot.title = element_text(size=2
4, hjust = 0.5), axis.title.x = element_text(size=24, vjust = -0.5, hjust = 0.71, color ='dimgr
ay')
    theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hjust
 = 0.5), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = element_text(size=
34), axis.title.y = element_text(size=34), axis.ticks = element_blank())+
      theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(4, 0.6)
            ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
            ,axis.line=element_line(colour=NA), axis.ticks = element_line(size=1.5, color ='dim
gray'), axis.ticks.length = unit(7, "pt")
            ,legend.key = element_rect(fill = NA, color = NA))+
        guides(colour = guide_legend(override.aes = list(size=5)))+theme(panel.grid.major=eleme
nt_blank(),panel.grid.minor=element_blank())
```



Hide

```
# dev.off()

ggsave(paste("./figure3/3main_feast.pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

```
Saving 12.9 x 8 in image
```

Hide

```
# fig3i = fig3i + 1
```

Hide

```
one_fold_change_record <- function(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na, n_top=30){

    meta_fil_config1_sub <- meta_fil_config1_na[,c("SRA_Sample", "Previous_sra", "Donor_sra")]


    ##filter top30 abundance genus
    # n_top = 30
    genus_sum <- rowSums(L6_rela_fil_sAg_remove_a1[,meta_fil_config1_sub[,c('SRA_Sample')]])
    top_index = order(genus_sum,decreasing=TRUE)[1:n_top]

    one_fold_change <- apply(meta_fil_config1_sub, 1, function(x, L6_rela_fil_sAg_remove_a1){
        index <- match(x, colnames(L6_rela_fil_sAg_remove_a1))
        index_L6 <- L6_rela_fil_sAg_remove_a1[,index]

        index_fc<- apply(index_L6, 1, function(y, x){
            out_p <- log((y[1]/y[2])+1);
            out_d <- log((y[1]/y[3])+1);
            c(out_p, out_d, y[1]);
            return(out_p - out_d)}, x=x)

        return(index_fc)
    #
    #     # quadrant1 <- sum(index_fc[3,index_fc[1,] > index_fc[2,] ])/1
    #     # quadrant2 <- sum(index_fc[3,index_fc[1,] < index_fc[2,] ])/1#00000
    #     quadrant1 <- mean((index_fc[1, index_fc[1,] > index_fc[2,]] - index_fc[2, index_fc[1,] > index_fc[2,]])*index_fc[3, index_fc[1,] > index_fc[2,]], trim=0.05)*length(index_fc[3,index_fc[1,] > index_fc[2,]])##
    #     quadrant2 <- mean((index_fc[2, index_fc[1,] < index_fc[2,]] - index_fc[1, index_fc[1,] < index_fc[2,]])*index_fc[3, index_fc[1,] < index_fc[2,]], trim=0.05)*length(index_fc[3,index_fc[1,] < index_fc[2,]])
    #     -log(quadrant2/quadrant1)
    }, L6_rela_fil_sAg_remove_a1[top_index,])

    return(one_fold_change)
    #
    # fc_coin <- as.data.frame(cbind(as.numeric(one_fold_change), (meta_fil_config1_na$postfmt_symptoms), (meta_fil_config1_na$PRJ)))
    # colnames(fc_coin)<- c('fc', 'group', 'prj')
    # rownames(fc_coin) <- (meta_fil_config1_na$SRA_Sample)
    # fc_coin$fc <- as.numeric(as.character(fc_coin$fc))
    #
    # prj_list <- fc_coin$prj
    # list <- NULL
    # for(i in unique(prj_list)){
    #     if (length(prj_list[prj_list %in% i]) > 2){
    #         list <- c(list, i)
    #     }
    # }
    # fc_coin_f <- fc_coin[fc_coin$prj %in% list,]
    #
    # pval<-wilcox_test(fc ~ group | prj, fc_coin_f)
    # pval
    # return(list('fc_ratio'=fc_coin, 'ratio_p'=pval))


}
```

```
row_select_sum <- function(dat_01){
    sum_out <- apply(dat_01, 1, function(x){
        hs = sum(x[x>0])
        ls = sum(x[x<0])
        if(hs >= -ls){
            return(hs)
        }else{
            return(ls)
        }
    })
    return(sum_out)
}
```

Hide

```
get_stable_engra <- function(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na_tmp, rate=0.5, n_top=20){


    genus_fc_per_person <- one_fold_change_record(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na_tmp, n_top)#meta_fil_config1_na$pre_entro %in% c('before1')

    dim(genus_fc_per_person)

    genus_fc_per_person_01 <- genus_fc_per_person


    genus_fc_per_person_01 <- ifelse(genus_fc_per_person > 0, 1, ifelse(genus_fc_per_person < -0, -1, 0))


    genus_fc_per_person_01_len <- data.frame(cbind(sapply(as.character(rownames(genus_fc_per_person_01)), simp_names), row_select_sum(genus_fc_per_person_01)), stringsAsFactors = FALSE)
    genus_fc_per_person_01_len[,2] <- as.numeric(genus_fc_per_person_01_len[,2])

    genus_fc_per_person_01_len_stable <- (genus_fc_per_person_01_len[((genus_fc_per_person_01_len[,2]) > rate*dim(genus_fc_per_person)[2]) | (genus_fc_per_person_01_len[,2] < -rate*dim(genus_fc_per_person)[2]),])

    print(length(genus_fc_per_person_01_len_stable[genus_fc_per_person_01_len_stable[,2] > 0, 1]))
    print(length(genus_fc_per_person_01_len_stable[genus_fc_per_person_01_len_stable[,2] < 0, 1]))

    return(genus_fc_per_person_01_len_stable)
}
```

Hide

```
all_stable <- get_stable_engra(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[,], 0.1)
```

```
[1] 15
[1] 5
```

Hide

```
before1_stable <- get_stable_engra(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c('before1'),], 0.1)
```

```
[1] 15
[1] 5
```

Hide

```
before2_stable <- get_stable_engra(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c('before2'),], 0.1)
```

```
[1] 12
[1] 8
```

Hide

```
res_stable <- get_stable_engra(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('response'),], 0.1, n_top=228)
```

```
[1] 116
[1] 79
```

Hide

```
fail_stable <- get_stable_engra(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('failure'),], 0.1, n_top=228)
```

```
[1] 106
[1] 93
```

Hide

```
mean(colSums(L6_rela_fil_sAg_remove_a1[rownames(res_stable)[res_stable$X2 > 0], meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('response'), 'SRA_Sample']])/colSums(L6_rela_fil_sAg_remove_a1[, meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('response'), 'SRA_Sample']]))
```

```
[1] 0.8367885
```

Hide

```
mean(colSums(L6_rela_fil_sAg_remove_a1[rownames(fail_stable)[fail_stable$X2 < 0], meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('response'), 'SRA_Sample']])/colSums(L6_rela_fil_sAg_remove_a1[, meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('response'), 'SRA_Sample']]))
```

```
[1] 0.1583507
```

Hide

```
mean(colSums(L6_rela_fil_sAg_remove_a1[rownames(res_stable)[res_stable$X2 > 0], meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('failure'), 'SRA_Sample']])/colSums(L6_rela_fil_sAg_remove_a1[, meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('failure'), 'SRA_Sample']]))
```

```
[1] 0.6826762
```

Hide

```
mean(colSums(L6_rela_fil_sAg_remove_a1[rownames(fail_stable)[fail_stable$X2 < 0], meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('failure'), 'SRA_Sample']])/colSums(L6_rela_fil_sAg_remove_a1[, meta_fil_config1_na[meta_fil_config1_na$postfmt_symptoms %in% c('failure'), 'SRA_Sample']]))
```

```
[1] 0.3159871
```

Hide

```
out <- NULL

for(i in 1:208){
    out <- rbind( out, c(simp_names(rownames(L6_rela_fil_sAg_remove_a1)[i]), cor(L6_rela_fil_sAg_remove_a1['k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Peptostreptococcaceae;g__[Clostridium]', meta_fil_config1_na$SRA_Sample], L6_rela_fil_sAg_remove_a1[i, meta_fil_config1_na$Donor_sra], method = 'spearman'), rownames(L6_rela_fil_sAg_remove_a1)[i]))

}
```

```
丣牠攷狜  丣牉攷狂Ï挼牠攷犄丣牠攷狜  丣牉攷狂Ï挼牠攷犄丣牠攷狜  丣牉攷狂Ï挼牠攷犄丣牠攷狜  丣牉攷狂Ï挼牠攷犄
```

Hide

```
out <- data.frame(out)
out$X2 <- as.numeric(as.character(out$X2))
```

Hide

```
write.csv(out[out$X2 < -0.1, c('X3', 'X2')], file = 'tmp1_donor.csv')
```

Hide

```
genus_fc_per_person <- one_fold_change_record(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[meta_fil_config1_na$Diease1 %in% c('CDI'),])#meta_fil_config1_na$pre_entro %in% c('before1')

dim(genus_fc_per_person)
```

```
[1]  30 188
```

Hide

```
cor_out <- apply(genus_fc_per_person, 1, function(x){
    c( 0, cor(((x)),  as.numeric(as.factor(meta_fil_config1_na[meta_fil_config1_na$Diease1 %in%
'CDI', 'postfmt_symptoms'])), method = 'spearman'))})
cor_out_dat <- data.frame(t(cor_out), stringsAsFactors = FALSE)
```

Hide

```
genus_fc_per_person <- one_fold_change_record(L6_rela_fil_sAg_remove_a1, meta_fil_config1_na[me
ta_fil_config1_na$Diease1 %in% c('CD', 'UC'),])#meta_fil_config1_na$pre_entro %in% c('before1')

dim(genus_fc_per_person)
```

```
[1] 30 98
```

Hide

```
cor_out <- apply(genus_fc_per_person, 1, function(x){
    c( 0, cor(((x)),  as.numeric(as.factor(meta_fil_config1_na[meta_fil_config1_na$Diease1 %in%
c('CD', 'UC'), 'postfmt_symptoms'])), method = 'spearman'))})
cor_out_dat <- data.frame(t(cor_out), stringsAsFactors = FALSE)
```

Hide

```
trim_r = 0.04
quntile_fold_change_1 <- function(L6_rela_fil_sAg_remove_a1, meta_fil_config1_1, pre_don, quan)
{ #"Previous_sra"
    sid_SRA_Sample <- meta_fil_config1_1$SRA_Sample
    sid_pre_don<- meta_fil_config1_1[, pre_don]

    # colnames(L6_rela_fil_sAg_remove_a1)
    index_SRA_Sample <- match(sid_SRA_Sample, colnames(L6_rela_fil_sAg_remove_a1))
    index_pre_don <- match(sid_pre_don, colnames(L6_rela_fil_sAg_remove_a1))

    quntile_fold_change <- apply(L6_rela_fil_sAg_remove_a1, 1, function(x, index_SRA_Sample, in
dex_pre_don){
        fold <- mean(x[index_SRA_Sample], trim=trim_r) / mean(x[index_pre_don], trim=trim_r)#me
an(quantile((x[index_SRA_Sample]), quan)) / mean(quantile((x[index_pre_don]), quan))
        mean_fold<- (log((fold) + 1));
        # mean_fold<- mean(log((quantile((x[index_pre_don]), quan) / quantile((x[index_SRA_Samp
le]), quan)) + 1));

        ##mean
        c(fold, mean_fold, mean(x[index_SRA_Sample], trim=trim_r))##
    }, index_SRA_Sample=index_SRA_Sample, index_pre_don=index_pre_don)

    # meta_fil_config1[, 'SRA_Sample']
    # apply(L6_rela_fil_sAg_remove_a1, 1, function(x){
    #     c(mean(x[, meta_fil_config1$SRA_Sample]), mean(x[, meta_fil_config1[,pre_don]]))
    # })
    t(quntile_fold_change)
}
```

Hide

```
##genera engraft validate
donor_before_after_color <- c("#9F452A", "#4E86C6", "#235E27")
L6_rela_fil_sAg_remove_a1_simp <- L6_rela_fil_sAg_remove_a1
L6_rela_fil_sAg_remove_a1_simp_name <- sapply(as.character(rownames(L6_rela_fil_sAg_remove_a1)), simp_names)
rownames(L6_rela_fil_sAg_remove_a1_simp) <- c(L6_rela_fil_sAg_remove_a1_simp_name)
```

Hide

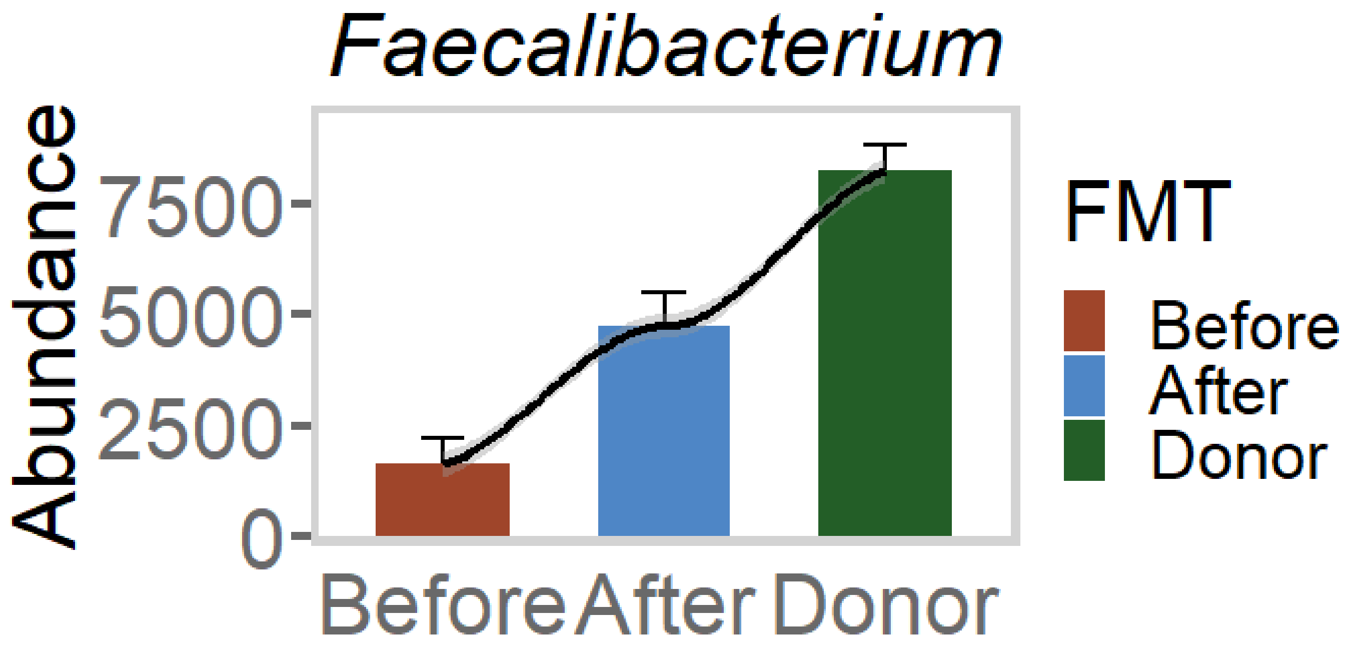
```
library(reshape2)
library(ggplot2)
library(dplyr)
trim_r=0.0
select_plot_genus_engra <- function(search_genus, sysptom_engra, disease=c('CDI', 'UC', 'CD'), meta_fil_config1_fil=meta_fil_config1){
     searched_abundance <- L6_rela_fil_sAg_remove_a1_simp[search_genus,]

     ###build two column selected metafile
     tmp_3column_before1 <- meta_fil_config1[(meta_fil_config1$postfmt_symptoms %in% c(sysptom_engra)) & (meta_fil_config1$Diease1 %in% c(disease)), c('Previous_sra', 'SRA_Sample', 'Donor_sra')]#meta_fil_config1_fil[,c('Previous_sra', 'SRA_Sample', 'Donor_sra')]#
    colnames(tmp_3column_before1) <- c('Before', 'After', 'Donor')

    # FMTstage_before1 <- melt(tmp_3column_before1, measure.vars = c('Before', 'After', 'Donor'))
    # FMTstage_before1$variable <- factor(FMTstage_before1$variable, levels = c('Before', 'After', 'Donor'))
    # ###select
    # FMTstage_abun_before1 <- cbind(FMTstage_before1, searched_abundance[FMTstage_before1$value])
    # colnames(FMTstage_abun_before1) <- c(colnames(FMTstage_before1), 'abun')
    #
    # stat_df <- FMTstage_abun_before1 %>% group_by(variable) %>% dplyr::summarise(mean_abun = mean((abun), trim=trim_r), sd_abun = 1.96*sd(abun)/sqrt(length(abun)))##mean
    FMTstage_before1 <- cbind(searched_abundance[tmp_3column_before1[,c('Before')]], searched_abundance[tmp_3column_before1[,c('After')]], searched_abundance[tmp_3column_before1[,c('Donor')]])
    colnames(FMTstage_before1) <- c('Before', 'After', 'Donor')

    tmp <- melt(FMTstage_before1)
    colnames(tmp)<-c('id', 'stage', 'value')
    tmp <- data.frame(tmp, stringsAsFactors = FALSE)
    tmp$stage <- factor(tmp$stage, levels=c('Before', 'After', 'Donor'))
    # tmp$stage <- as.numeric(tmp$stage)
    tmp$value <- as.numeric(tmp$value)

    stat_df <- cbind(c('Before', 'After', 'Donor'), c(mean((FMTstage_before1[,c('Before')]), trim=trim_r), mean((FMTstage_before1[,c('After')]), trim=trim_r), mean((FMTstage_before1[,c('Donor')]), trim=trim_r)), c(1.96*sd(FMTstage_before1[,c('Before')])/sqrt(length(FMTstage_before1[,c('Before')])), 1.96*sd(FMTstage_before1[,c('After')])/sqrt(length(FMTstage_before1[,c('After')])), 1.96*sd(FMTstage_before1[,c('Donor')])/sqrt(length(FMTstage_before1[,c('Donor')]))))
    colnames(stat_df) <- c('variable', 'mean_abun', 'sd_abun')
    stat_df <- data.frame(stat_df, stringsAsFactors = FALSE)
    stat_df$mean_abun <- as.numeric(as.character(stat_df$mean_abun))
    stat_df$sd_abun <- as.numeric(as.character(stat_df$sd_abun))
    stat_df$variable <- factor(stat_df$variable, levels = c('Before', 'After', 'Donor'))




    ggplot()+
         # geom_errorbar(aes(ymin=mean_abun-sd_abun, ymax=mean_abun+sd_abun), width=.2, position=position_dodge(.95), size=1)+
        geom_errorbar(aes(x = variable, y = mean_abun, ymin = mean_abun+sd_abun, ymax = mean_abun+sd_abun), data = stat_df,  width=.2, position=position_dodge(.95), size=1)+
    geom_linerange(aes(x = variable, y = mean_abun, ymin = mean_abun, ymax = mean_abun+sd_abun), data = stat_df,  size=1)+
    geom_bar(aes(x = variable, y = mean_abun, fill = variable), data = stat_df,  stat="identit
y", width = 0.6, alpha=1)+ #position = position_jitter(w = 0.35, h = 0.1), size=2,
    geom_smooth(aes(x=as.numeric(stage), y=value), data = tmp, color='black', level = 0.6, meth
od='loess', size=1.5)+
    # geom_boxplot(aes(x = variable, y = log(abun + 1)), FMTstage_abun_before1, color='black',
 alpha=0, size=0.8)+
    labs(x= c(''), y=c('Abundance'), title = c(search_genus))+
  #scale_colour_manual(name="FMT", values=c("#962E2B", "#4E86C6", "#4D9127", "#90908D", 'lightg
rey'))+#'#C77CFF', '#43AFC8',
  scale_fill_manual(name="FMT", values=c(donor_before_after_color, "#90908D", 'lightgrey'))+
        scale_y_continuous(expand = expansion(mult =c(0.02, 0.1)))+
      theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hjus
t = 0.5, face='italic'), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = el
ement_text(size=34), axis.title.y = element_text(size=34), axis.ticks= element_blank(), axis.ti
cks.y = element_line(size=1.5, color ='dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 0.62, legend.background=element_blank()#, legend.position=c(1.75, 0.6)
        ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
        ,axis.line=element_line(colour=NA)
        ,legend.key = element_rect(fill = NA, color = NA))+
    guides(colour = guide_legend(override.aes = list(size=5)))+theme(panel.grid.major=element_b
lank(),panel.grid.minor=element_blank())
}
```

Hide

```
plot_engra <- function(response, donor, theshold, disease=c('CDI', 'UC', 'CD')){
    # response = 'response'
    meta_fil_config1_fil <- meta_fil_config1[meta_fil_config1$postfmt_symptoms %in% c(response)
& meta_fil_config1$Diease1 %in% c(disease),]
    if(donor==1){
        pre_don_fc <- quntile_fold_change_1(L6_rela_fil_sAg_remove_a1_simp, meta_fil_config1_fi
l, "Previous_sra", quan)
    }else{
        pre_don_fc <- quntile_fold_change_1(L6_rela_fil_sAg_remove_a1_simp,meta_fil_config1_fi
l, "Donor_sra", quan)
    }

    engra_tmp <- pre_don_fc[pre_don_fc[,2] > theshold,]
    print(engra_tmp[,])
    engra_tmp_names <- rownames(engra_tmp)

    pi_engra <- list()
    for(i in engra_tmp_names){
        pi_engra[[i]] <- select_plot_genus_engra(i, response, disease, meta_fil_config1_fil)
    }

    pi_engra
}
```

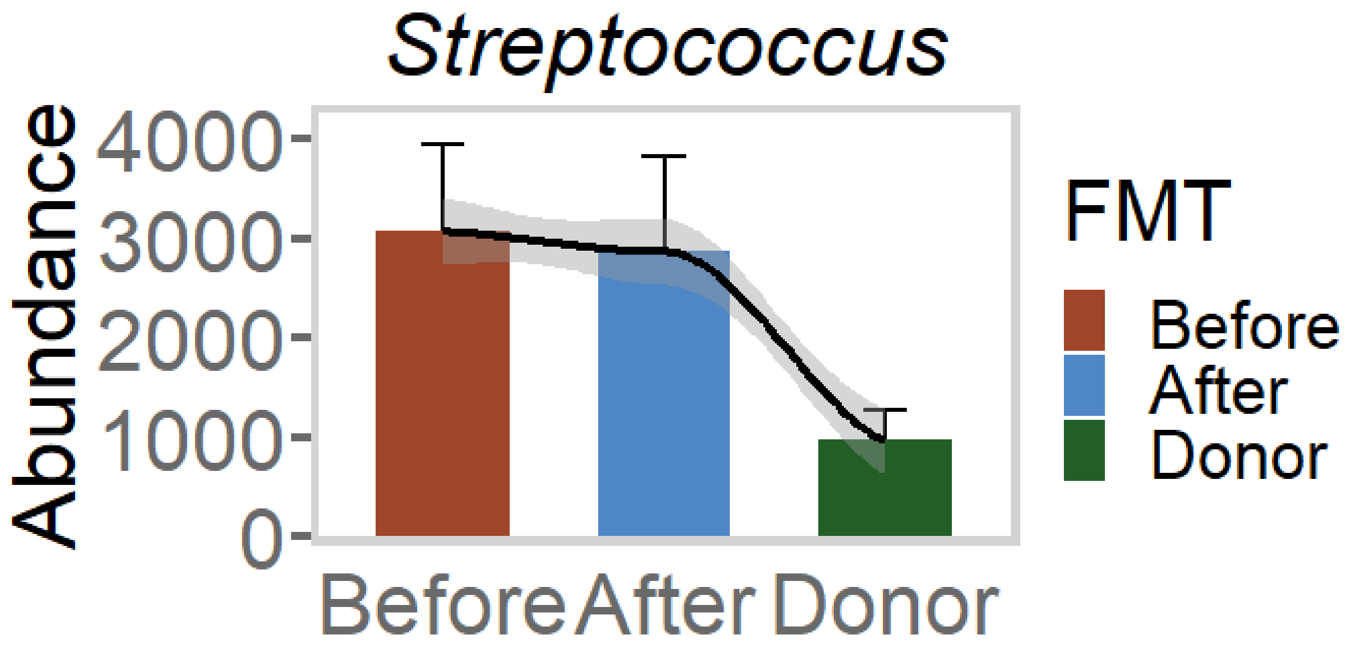
Hide

```
# pdf('figure3/engreft_genu_disease.pdf', width = 8)
select_plot_genus_engra('Faecalibacterium', c('failure', 'response'))
```



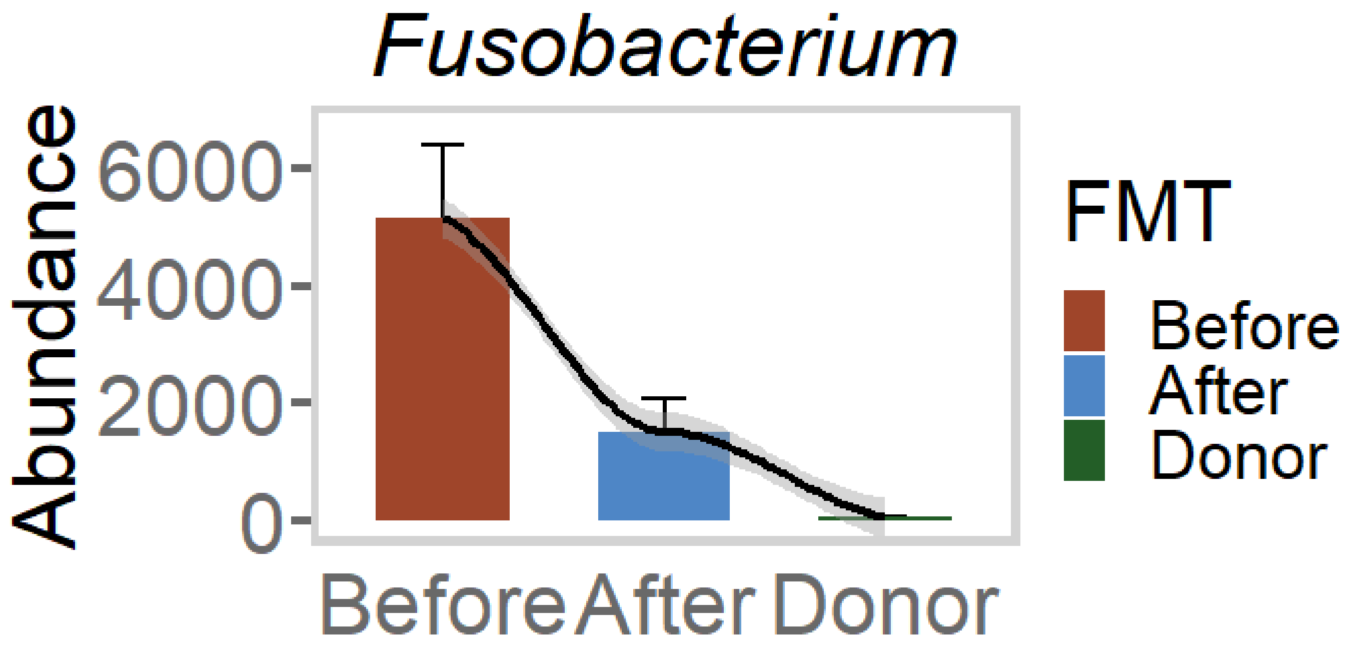
Hide

```
select_plot_genus_engra('Streptococcus', c('failure', 'response'))
```



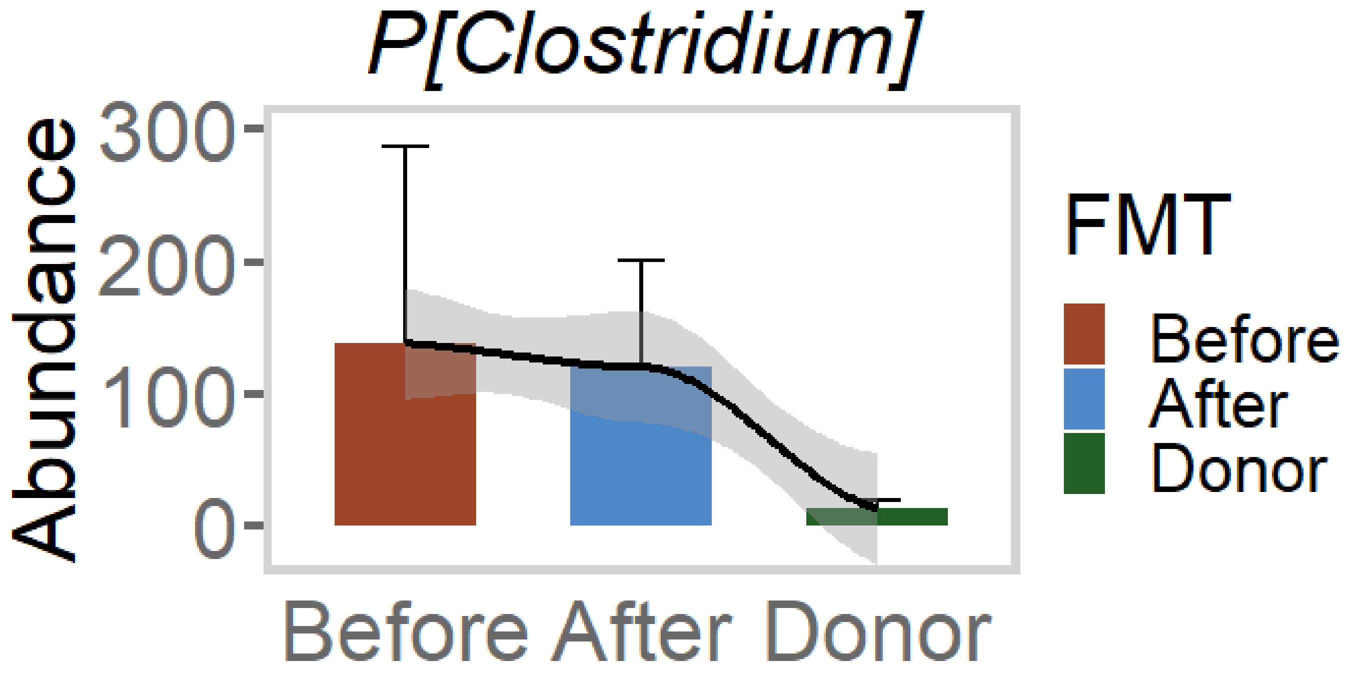
Hide

```
select_plot_genus_engra('Fusobacterium', c('failure', 'response'))
```



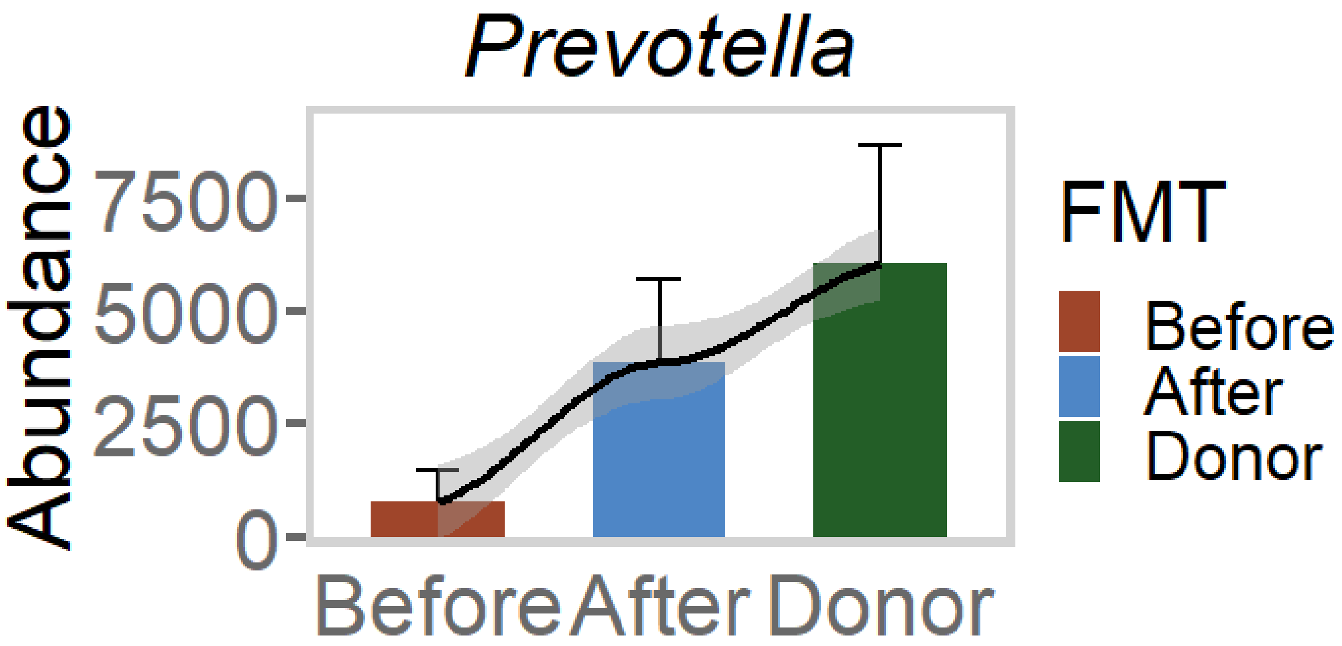
Hide

```
select_plot_genus_engra('P[Clostridium]', c('failure', 'response'), c('CDI'))
```



Hide

```
select_plot_genus_engra('Prevotella', c('failure', 'response'), c('UC', 'CD'))
```

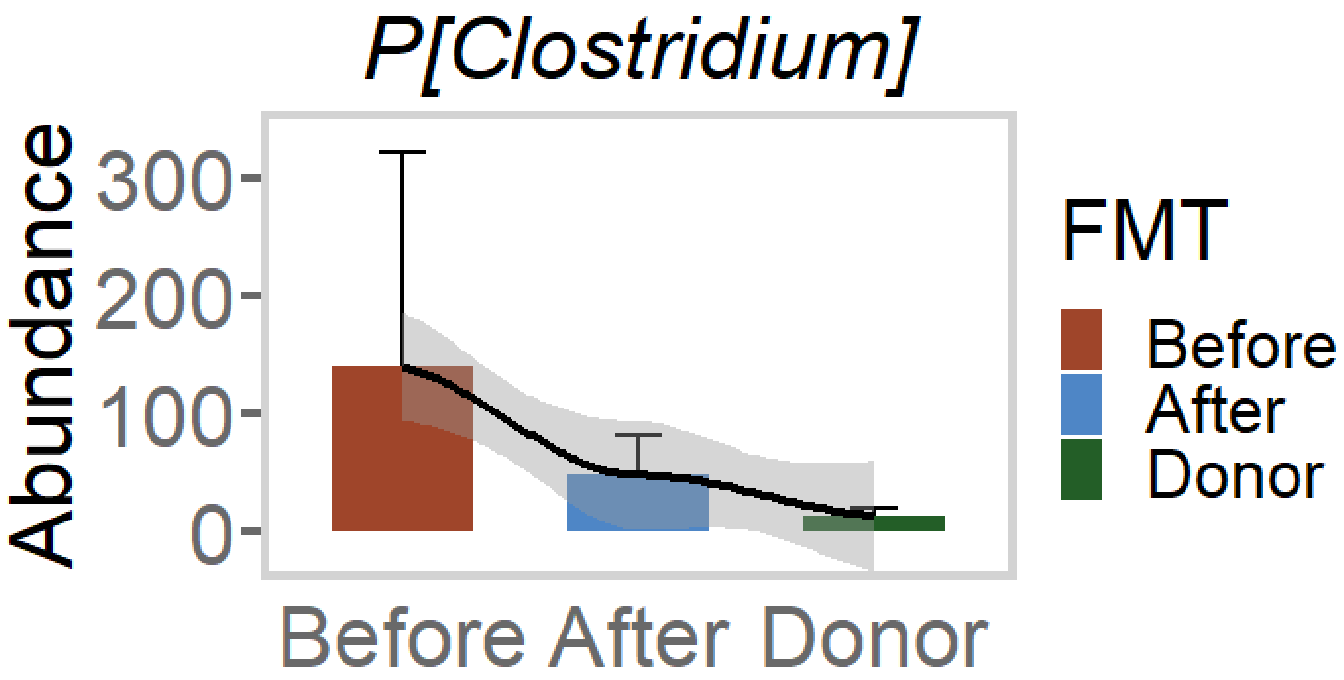


Hide

```
# dev.off()
```

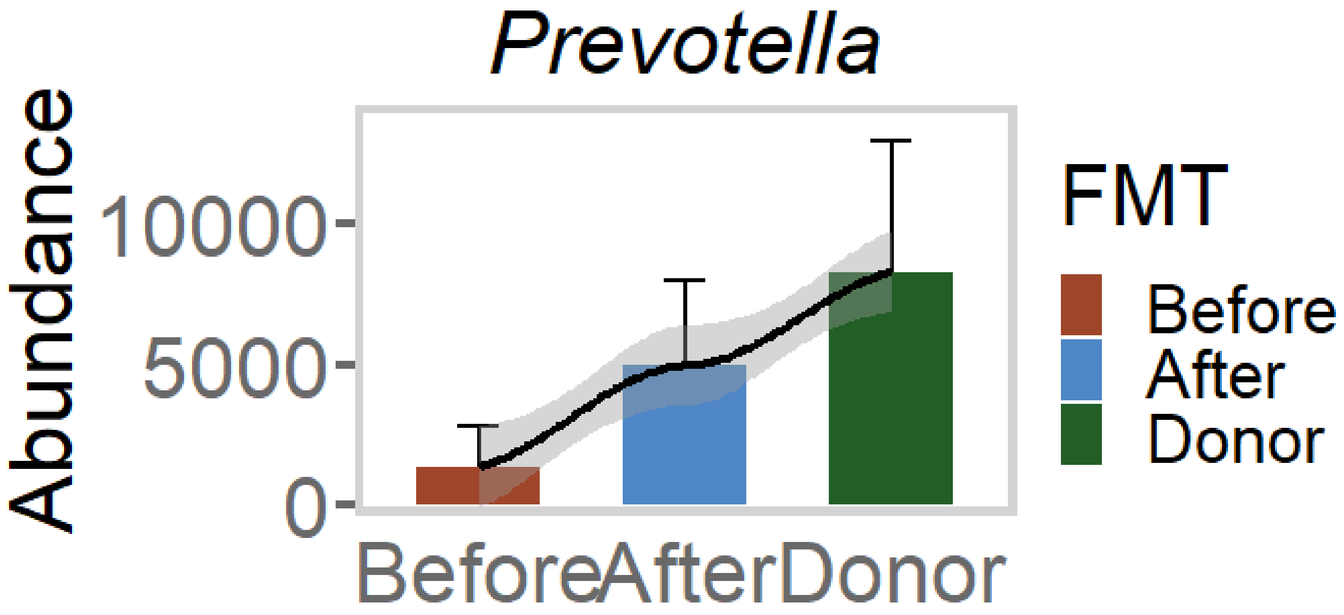
Hide

```
# pdf('figure3/engreft_genu_disease.pdf', width = 8)

select_plot_genus_engra('P[Clostridium]', 'response', c('CDI'))
```



Hide

```
select_plot_genus_engra('Prevotella', 'response', c('UC', 'CD'))
```



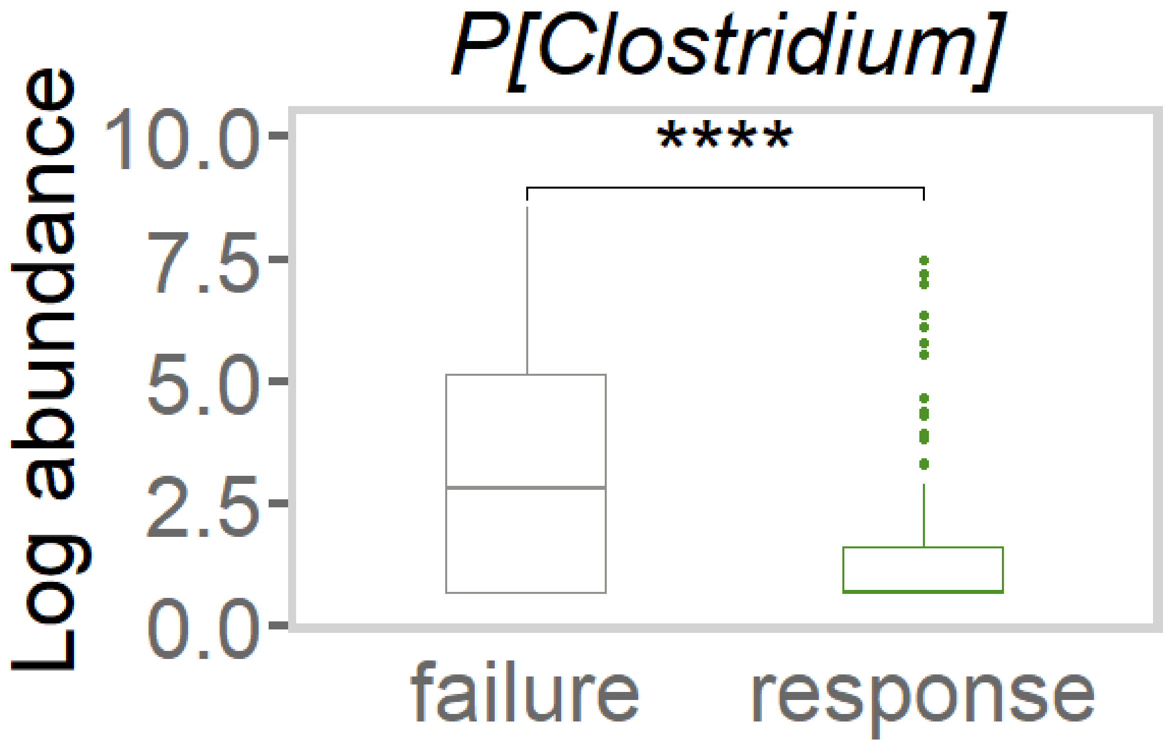
Hide

```
# dev.off()
```

Hide

```
##'P[Clostridium]'

search_genus = 'P[Clostridium]'
searched_abundance <- L6_rela_fil_sAg_remove_a1_simp[search_genus,]

after_cdi <- meta_fil_config1[(meta_fil_config1$Diease1 %in% c('CDI')) & (meta_fil_config1$post
fmt_symptoms %in% c('failure', 'response')), c( 'SRA_Sample', 'postfmt_symptoms')]
plot_cdi <- cbind(searched_abundance[after_cdi$SRA_Sample], after_cdi)
colnames(plot_cdi) <- c('abun', 'SRA_Sample', 'postfmt_symptoms')

plot_cdi$abun <- log(plot_cdi$abun + 1)

ggboxplot(plot_cdi, x = 'postfmt_symptoms', y = 'abun', color='postfmt_symptoms', width=0.4)+
# (, aes(x = postfmt_symptoms, y = log(abun+1), color=postfmt_symptoms))+
    # geom_boxplot(width=0.4)+
    stat_compare_means(comparisons = list(c('failure', 'response')), method = 'wilcox.test', la
bel = "p.signif", size=12, label.x = 1.4)+
         # geom_errorbar(aes(ymin=mean_abun-sd_abun, ymax=mean_abun+sd_abun), width=.2, positio
n=position_dodge(.95), size=1)+
    #     geom_errorbar(aes(ymin = mean_abun+sd_abun, ymax = mean_abun+sd_abun), width=.2, posi
tion=position_dodge(.95), size=1)+
    # geom_linerange(aes(ymin = mean_abun, ymax = mean_abun+sd_abun), size=1)+
    # geom_bar(aes(fill = variable), stat="identity", width = 0.6, alpha=1)+ #position = positi
on_jitter(w = 0.35, h = 0.1), size=2,
    # geom_boxplot(aes(x = variable, y = log(abun + 1)), FMTstage_abun_before1, color='black',
 alpha=0, size=0.8)+
    labs(x= c(''), y=c('Log abundance'), title = c(search_genus))+
  scale_colour_manual(name="FMT", values=c( "#90908D", "#4D9127", "#962E2B", "#4E86C6", 'lightg
rey'))+#'#C77CFF', '#43AFC8',
  scale_fill_manual(name="FMT", values=c(donor_before_after_color, "#90908D", 'lightgrey'))+
        scale_y_continuous(expand = expansion(mult =c(0.1, 0.2)))+
      theme(text=element_text(family ="sans", size=32), plot.title = element_text(size=34, hjus
t = 0.5, face='italic'), axis.text = element_text(size=32, color ='dimgray'), axis.title.x = el
ement_text(size=34), axis.title.y = element_text(size=34), axis.ticks= element_blank(), axis.ti
cks.y = element_line(size=1.5, color ='dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 0.6, legend.background=element_blank(), legend.position=c(1.75, 0.6)
        ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
        ,axis.line=element_line(colour=NA)
        ,legend.key = element_rect(fill = NA, color = NA))+
    guides(colour = guide_legend(override.aes = list(size=5)))+theme(panel.grid.major=element_b
lank(),panel.grid.minor=element_blank())

ggsave(paste("./figure3/cdi_clostri.pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

```
Saving 7.29 x 4.5 in image
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.